# A $\Delta^0_2$ set with barely $\Sigma^0_2$ degree*

Rod Downey, Geoffrey LaForte, Steffen Lempp

**Abstract**

We construct a $\Delta^0_2$ degree which fails to be computably enumerable in any computably enumerable set strictly below $\emptyset'$.

## 1 Introduction

The lion's share of effort in classical computability theory over the last fifty years has been directed toward the study of relative *computability*. This paper is concerned with another, more neglected, yet still fundamental, notion of classical computability theory, namely that of relative *enumerability*.

Specifically, we ask questions concerning the relationship between sets $A$ and $B$ when $A$ is computably enumerable using $B$ as an oracle. For example, given a set $B$, we might ask what properties the class of sets $A$ which are c.e. relative to $B$ has. Conversely, for fixed $A$, we might wonder which degrees contain sets relative to which $A$ can be computably enumerated. In the present paper, our particular concern is with this latter type of question. We study $\Sigma^0_2$ sets and degrees, and their relations with the computably enumerable sets from which these more complex sets can themselves be computably enumerated.

By the Sacks Jump Theorem [3], given any noncomputable, computably enumerable $C$, any $\Sigma^0_2$ set with degree at least as great as $\mathbf{0}'$ has the same degree as the jump of some $B$ such that $C \not\leq_T B$. Clearly, any such $B$ must itself be incomplete. Thus we have the weaker fact that any $\Sigma^0_2$ set with degree at least as great as $\mathbf{0}'$ has $\Sigma^B_1$-degree for some incomplete c.e. $B$. Of course, any $\Sigma^0_2$ set is itself relatively computably enumerable in $K$, and so has $\Sigma^K_1$-degree. The natural question in this context is whether or not any $\Sigma^0_2$ degree requires $K$ in order to witness that it is $\Sigma_2$, in other words, whether or not there exists a $\Sigma^0_2$ set $A$ such that whenever $A$ has $\Sigma^W_1$ degree for some c.e. set $W$, $W$ must be complete, a situation we describe by calling $A$ *barely* $\Sigma^0_2$.

Any c.e. set is computably enumerable relative to $\emptyset$, and, by an unpublished result of Lachlan, any 2-c.e. set is 2-CEA, that is, relatively c.e. in some c.e. set below it. Thus, any $\Sigma^0_2$ degree which requires the full power of $K$ for an enumeration must be of at least properly 3-c.e. degree. Since, by Arslanov-LaForte-Slaman, [1], no properly 3-c.e. degree can contain a set which is c.e. relative to a

*to appear, *Journal of Symbolic Logic*

c.e. set below it, it is not unreasonable to look among such degrees for $\Sigma_2^0$ ones that require $K$ to be enumerated. Our result here shows that we can find such barely $\Sigma_2^0$ degrees in this otherwise simple realm:

**Theorem** 1 **:** There exists a 3-c.e. set $A$ such that for all c.e. sets $W$, $A$ has $\Sigma_1^W$ degree if and only if $W \equiv_T \emptyset'$.

Thus, in general, a $\Sigma_2^0$ set need not be of $\Sigma_1^B$ degree for any incomplete c.e. set $B$.

We remark that if we consider sets themselves, rather than the degrees of sets, it is easy to see that $\overline{K}$ cannot be $\Sigma_1^B$ for any c.e. $B <_T \emptyset'$. In fact, any set to which $\overline{K}$ is $m$-reducible must have this same property, for instance, $\emptyset''$. The Noninversion Theorem of Shore, [4], gives us another means of exhibiting such a phenomenon. By this result, there exist two $\Sigma_2^0$ sets $U$ and $V$ such that $U \oplus V <_T \emptyset''$, and $U$ and $V$ cannot both have $\Sigma_1^B$ degree for any $B <_T \emptyset'$. Hence $U \oplus V$ cannot be $\Sigma_1^B$ for any such $B$. It does not seem to follow directly from this construction that $\overline{K}$ is computable from the set $U \oplus V$ so constructed.

Of course, these examples and even the $\Delta_2^0$ degree we construct leave almost completely open the more general question of exactly which $\Sigma_2^0$ sets below $\emptyset''$ have this property, as well as the analogous question about degrees. In particular, we have not constructed such a set incomparable with $\emptyset'$, although it seems natural to conjecture that there are such sets.

## 2 Theorem and general plan

**Theorem 1.** *There exists a* 3*-c.e. set* $A$ *such that for all c.e. sets* $W$*,* $A$ *has* $\Sigma_1^W$ *degree if and only if* $W \equiv_T \emptyset'$*.*

For every pair of partial computable functionals $\Phi$ and $\Psi$, and natural numbers $e$ and $l$, we must satisfy the requirement

$$\mathrm{R}_{\Phi,\Psi,e,l} \; : \; (\Phi(W_l^{W_e}) = A \text{ and } \Psi(A) = W_l^{W_e}) \Longrightarrow K \leq_T W_e.$$

We intend to satisfy $\mathrm{R}_{\Phi,\Psi,e,l}$ by using an $\omega$-sequence of functionals, $\Gamma_n(W_e)$, together with a "backup" functional $\Gamma_\infty(W_e)$. In order make our notation less cumbersome, we generally refer to these functionals as just $\Gamma_n$ or $\Gamma_i nfty$ in what follows, avoiding explicit reference to $W_e$ when this is clear from the context. $\Gamma_\infty$ will be defined on $n$ each time there is an uncorrectable failure of $\Gamma_n$ to compute a value of $K$. We implement this strategy by defining a length-of-agreement function approximating the truth of $\Phi(W_l^{W_e}) = A$ and $\Psi(A) = W_l^{W_e}$. Assuming that the condition holds, and we are currently defining some $\Gamma_n$, we split $\mathrm{R}_{\Phi,\Psi,e,l}$ into subrequirements

$$\mathrm{R}_{\Phi,\Psi,e,l}(i) \; : \; \Gamma_n(W_e; i) = K(i) \text{ or } \Gamma_\infty(W_e; n) = K(n),$$

which are allowed to act at $\mathrm{R}_{\Phi,\Psi,e,l}$-expansionary stages.

Our basic strategy has two parts: an attempt, possibly without success, to correctly define $\Gamma_n(i)$, followed by a definition of $\Gamma_\infty(n)$ which is guaranteed to succeed. By a slight abuse of standard notation, we let $\phi_l(W_e;x)[s]$ be the maximum of the uses of all computations $\phi_l(W_e;y)[s]$ for which $y \leq x$ and $y \in W_l^{W_e}[s]$. At stage $s$ we assign an attacker $a \notin A[s]$ to our subrequirement, set the use of $\Gamma_n(i)[s]$ to be $s > \phi_l(W_e;\phi(W_l^{W_e};a))[s]$, and restrain $A$ below $\psi(\phi(a))$. When $i$ enters $K$, we enumerate $a$ into $A$, forcing a change on $\phi(W_l^{W_e};a)[s]$ at some later stage $t$. If this is caused by an element leaving $W_l^{W_e}[s]$, this involves a change in $W_e$ below $\gamma_n(i)$. Otherwise, some new element $x$ is added to $W_l^{W_e}[t]$. We then set $\gamma_\infty(n) = \phi_l(x)$, the use from $W_e$ by which $x$ is an element of $W_l^{W_e}$. Now, by restraining $A$ appropriately, we can force changes at will on $W_e$ by forcing $x$ in and out of $W_l^{W_e}$ through changes on $A(a)$. This involves only at most one further change on $A$ to keep $\Gamma_\infty(W_e;n)$ correct up to $n$, so $A$ remains $\Delta_2^0$ as required. In fact, from the point of view of this requirement in isolation, $A$ appears to be 2-c.e. Avoiding injury to other (higher-priority) requirements, however, involves restoring the value of $A(a)$ at the stage at which these requirements acted to set their original use, so there is at least an apparent potential for infinitely many changes to be required on $a$ through a cascading effect caused by sequences of restorations. It is the avoidance of this that is the fundamental obstacle to achieving the proof.

The fact that we need a $\mathbf{0}'''$-priority arrangement to organize our construction arises naturally from the purely local problem of infinite injury to the use $\phi(W_l^{W_e};a)$ through changes in $W_l^{W_e}$. From the standpoint of the overall requirement $\mathrm{R} = \mathrm{R}_{\Phi,\Psi,e,l}$, there are three possible outcomes. Either there are only finitely many expansionary stages, or some $\Gamma_n$ succeeds in computing $K$, or $\Gamma_\infty$ is total. In the usual $\mathbf{0}'''$ manner, infinite injury to some use can occur below each of the two infinitary outcomes and thereby deny the truth of these higher-level approximations. Before giving the full construction, we describe the basic module in more detail, and then discuss the intuition for the priority arrangement.

## 2.1 The basic module for $\mathrm{R}_{\Phi,\Psi,e,l}$

Let $\mathrm{R} = \mathrm{R}_{\Phi,\Psi,e,l}$, and assume some good approximation $l^{\mathrm{R}}(s) = l(s)$ has been defined with

$$l(s) = \max(\{\, x : \big(\Phi(W_l^{W_e};x) = A(x), \text{ and } \Psi(A) \upharpoonright \phi(W_l^{W_e};x) = W_l^{W_e} \upharpoonright \phi(W_l^{W_e};x)\big)[s] \,\}).$$

We consider the action taken for $\mathrm{R}(i)$, attempting either to keep $\Gamma_n(i) = K(i)$ or $\Gamma(n) = \Gamma_\infty(n) = K(n)$. The following is the basic module for action at R-expansionary stages, beginning with a stage $s_0$. We need only consider the case $i \notin K[s_0]$

**1.** Choose $a = a_n(i) \notin A[s_0]$.

**2.** Wait for the next R-expansionary stage $s_1$ with $l(s_1) > a$. If $i \in K[s_1]$, we merely set $\gamma_n(i) = s_1$ and $\Gamma_n(i) = 1$. If $i \notin K[s_1]$, then we restrain $A$ on $\max(\{\,\psi(A;y)[s_1] \,:\, y < \phi(W_l^{W_e};a)[s_1]\,\})$. Set $\gamma_n(i)[s_1] = \max(\{\,\phi_l(y)[s_1] \,:\, y \in W_l^{W_e} \restriction \phi(a)[s_1]\,\}$.

**3.** Wait for an R-expansionary stage $s_2$ such that $i \in K[s_2]$.

**A.** If $W_e$ changes on $\gamma_n(i)$ at some stage $s > s_1$ before $i$ enters $K$, then reset $\gamma_n(i)[s+1]$ as in **2**.

**B.** If $i \in K[s_2]$ (and $\gamma_n(i)[s_2]\downarrow$), then add $a_n(i)$ to $A[s_2+1]$, and restrain $A$ on $\max(\{\,\psi(A;y)[s_2] \,:\, y < \phi(W_l^{W_e};a)[s_2]\,\})$. Go to **4**.

**4.** At the next R-expansionary stage $s_3$, there are two possibilities.

**A.** $\Gamma_n(i)[s_3]\uparrow$. Then set $\Gamma_n(i) = 1$, and reset $\gamma_n(i) = s_3$, permanently. [In the full construction, we also remove $a_n(i)$ from $A[s_3+1]$ in order to avoid infinite injury to lower priority requirements.]

**B.** $\Gamma_n(i)[s_3]\downarrow$. This can only occur if there is some $y \in (W_l^{W_e}[s_3] - W_l^{W_e}[s_2])$. We fix $y'$ to be the greatest such number. Again, we only take significant action if $n \notin K[s_3]$; otherwise, we just set $\gamma(n) = s_3$ and $\Gamma(W_e;n) = 1$. If $n \notin K[s_3]$, however, we set $\gamma(n)[s_3+1] = \phi_l(W_e;y')[s_3]$, and restrain $A$ on $\max(\{\,\psi(A;y)[s_3] \,:\, y < \phi(W_l^{W_e};a)[s_3]\,\})$, as well as maintaining the previous restraint.

**5.** Wait for an R-expansionary stage $s_4$ such that $n \in K[s_4]$.

**A.** If $W_e$ changes on $\gamma(n)$ at stage $s > s_3$ before $n$ enters $K$, then reset $\gamma(n)[s+1]$ as in **4B**. Maintain restraint on $A$.

**B.** If $n \in K[s_4]$ (and $\gamma_n(i)[s_4]\downarrow$), remove $a_n(i)$ from $A[s_4+1]$. Go to **6**.

**6.** At the next R-expansionary stage, $s_5$, we must have $\Psi(A) \restriction \phi(a_n(i))[s_5] = \Psi(A) \restriction \phi(a_n(i))[s_2]$, hence $y' \notin W_l^{W_e}[s_5]$. Hence $\gamma(n)[s_5]\uparrow$. Then set $\Gamma(n) = 1$, and reset $\gamma(n) = s_5$, permanently. [Again, as in **4A**, in the full construction we add $a_n(i)$ back to $A[s_5+1]$ to avoid injury to other requirements.]

There are six possible outcomes for this strategy. Four finitary outcomes at **2**, **4A**, **4B**, and **6**, and two infinitary ones at **3A** and **5A**. Notice that if the infinitary outcomes occur infinitely often, then R is satisfied by diagonalization, since some number is counted by $\Psi(A)$ as an element of $W_l^{W_e}$, yet fails to actually be in $W_l^{W_e}$. We can initialize all lower-priority strategies when the action under **4B** above causes a significant shift in our overall strategy, since we give up $\Gamma_n$ entirely at this point. This means as well that we will never return to step **3** once the conditions of **4B** are met. Because we are in control of $A$, we have the authority to do this, since the number $y'$ will have to be an element of $W_l^{W_e}$ at any stage where $\Psi(A) = W_l^{W_e}$ appears correct. Thus this outcome initializes all strategies with lower priority than R itself which are guessing that one of the attempts at a non-backup functional will succeed; at which point we extend the "backup" functional $\Gamma_\infty$, and thereafter begin anew with the new non-backup attempt $\Gamma_{n+1}$. We therefore arrange the three possible outcomes for the overall strategy with highest priority outcome the totality of $\Gamma_\infty$ ("$\infty$"), to the right of which is the totality of some $\Gamma_n$ ("num"), and, finally, with lowest priority, the existence of only finitely many expansionary stages ("fin"). Below each of the two infinitary outcomes, lies a sequence of subrequirements $\mathrm{R}(i)$ each of which sets a restraint for the sake of preserving the computation tied to its witness. Each of these substrategies has natural outcomes $\infty$ and fin, depending on whether its restraint is increased infinitely often or not. It is worth pointing out here that because we restrain $A$ after action under **4B** is taken, we can never get another expansionary stage without the $y'$ referred to there being an element of $W_l^{W_e}$, so we never have to worry about $\Gamma(n)$ being incorrect: $\Gamma(n)$ can only fail to equal $K(n)$ if $\Psi(A) \neq W_l^{W_e}$.

## 2.2 Intuition for the priority arrangement

Since an overall requirement $\mathrm{R} = \mathrm{R}_{\Phi,\Psi,e,l}$ which actually defines a functional computing $K$ must eventually impose infinite restraint to ensure that its functional is defined everywhere, we must split it up into subrequirements. Once one of these subrequirements, say $\mathrm{R}(i)$, acts to define some $\Gamma_n^{\mathrm{R}}(i)$, we obviously cannot redefine our functional without an appropriate change in $W_e$. If we actually attack to correct this value, and fail, our strategy involves switching to the backup functional $\Gamma_\infty$ and giving up $\Gamma_n$ permanently. Thus if higher-priority strategies interfere with $\Gamma_n(i)$, this causes no real problem for the construction. On the other hand, no strategy below R can be allowed to interfere infinitely often with the functional $\Gamma_\infty$, since the whole point of having $\Gamma_\infty$ available is that it is guaranteed to succeed in computing $K$, if it is total. Notice that because we wish to construct a $\Delta_2^0$ set $A$, we cannot merely restore the previous $A$-state to protect the strategy whenever R is allowed to act. With infinitely many values of $\Gamma_\infty$ eventually defined, this could result in infinitely many changes on some element $a \notin A$. This is even more clear when one considers that, unless we take some explicit action to ensure that R is always depending on consistent initial segments of $A$, it is quite imaginable that the correctness of strategies tied to different values of $\Gamma_\infty$ depend on different $A$-states. In this case, we would not even be able to produce a consistent overall strategy for defining $\Gamma_\infty$. The considerations show that we really must restrain $A$ for the sake of $\Gamma_\infty$.

Consider the interference with $\Gamma_\infty$ that could be caused by some other strategy $\mathrm{S}(j)$ working for an overall requirement S. If S has higher global priority than R, then, using linking in the ordinary fashion, we can turn R off while $\mathrm{S}(j)$ is attacking and, after $\mathrm{S}(j)$'s attack, either restore the previous $A$-state, or initialize R. This is not a possibility in the case of S of lower global priority than R, but $\mathrm{S}(j)$ of higher local priority than $\mathrm{R}(i)$. This is the situation where S really must make a potentially permanent change in $A$ which can interfere with the correctness of even the backup strategy for $\Gamma_\infty^{\mathrm{R}}$ which is tied to $\mathrm{R}(i)$. In general, $\mathrm{S}(j)$ has an attacker which is smaller than the use on which the $\mathrm{R}(i)$-attacker is depending. When $\mathrm{S}(j)$ changes $A$ at stage $s+1$ this allows the particular element $y' \in W_l^{W_e}[s]$ which is used to define $\gamma_\infty^{\mathrm{R}}(n)$ to leave $W_l^{W_e}[s+1]$.

The natural action for S in this case is to first ask for an attack on R$(i)$, so as to clear the use of $\Gamma^{\mathrm{R}}_{\infty}$. In fact, this happens automatically in this case, since the R$(i)$-attacker is greater than S$(j)$-attacker. This will ensure that the R strategy for correcting $\Gamma^{\mathrm{R}}_{\infty}$ will always succeed, but it threatens to make it impossible for $\Gamma^{\mathrm{R}}_{\infty}(i)$ to converge in the limit, since there may in this case be infinitely many such lower-priority requirements which can affect the R$(i)$-strategy.

By using the standard convention that uses increase in the argument, after each successful attack at a $K$-true stage, the the active functional for the S-strategy will be completely undefined on any number which will later enter $K$. We can therefore be assured that all of the S-attackers still defined will never be used, since they will be assigned to substrategies for numbers that are not in $K$. Thus, if S waits to define new attackers until after the R$(i)$-strategy is reset, S can no longer interfere with R$(i)$, since it will only wish to change $A$ above the restraint imposed for the sake of R$(i)$. Notice that S is in this case a strategy based on the assumption that $\Gamma^{\mathrm{R}}_{\infty}$ is a total function. Because there are only finitely many requirements S between R and R$(i)$, this process must eventually stabilize, resulting in $\Gamma^{\mathrm{R}}_{\infty}(i)\downarrow$.

Coordinating the actions of the sequence $\mathrm{S}_1, \ldots, \mathrm{S}_m$ of strategies which can cause this interference is somewhat involved. Before giving the full details below, we should mention that it is useful to require the natural condition that if R has higher priority than S, and $i \leq j$, then R$(i)$ has higher priority than S$(j)$. In this way, the fact that $K$ has stabilized below $i$ ensures that eventually R$(i)$ will no longer be injured by S.

In order to explain in more detail the intuition for the interaction of various strategies in our proof, we require the basic notions about the tree method of Lachlan and Harrington in priority arguments, for which the reader should see [5], XIII. The simplest situation in which the complexity of our linking of strategies reveals itself is the following: suppose $\tau_0$, $\tau_1$, and $\tau_2$ are all master strategies with substrategies $\sigma_0$, $\sigma_1$, and $\sigma_2$, respectively, such that

$$\tau_0 \subset \tau_1 \subset \tau_2 \subset \sigma_2 \subset \sigma_1 \subset \sigma_0.$$

Let requirement $R_{\Phi_j,\Psi_j,e_j,l_j}$ be assigned to $\tau_j$, and $\mathrm{R}^{\tau_j}(k_j)$ be assigned to $\sigma_j$ with $k_2 < k_1 < k_0$. With so many requirements, the description is naturally rather involved. To simplify matters, we assume in this example that everything proceeds without involving a switch by any master strategy to its backup functional. Supposing, then, that each $\tau_j$ is making its $n_j$th attempt to define its non-backup functional, we write $a_j$ for $a^{\tau_j}_{n_j}(k_j)$, the current attacker assigned to $\sigma_j$. Because it may be helpful for the reader to refer back to this example when reading the formal construction below, we refer ahead in what follows to the cases of the formal construction from sections 3.2 and 3.3 using square brackets.

Suppose $\sigma_2$ wishes to attack in order to correct $\Gamma^{\tau_2}$'s value for $K(k_2)$ at stage $s$. To effect this, $\sigma_2$ links up to $\tau_2$ [section 3.3, case 8]. In general, $\psi_1(A;\phi_1(W^{W_{e_1}}_{l_1};a_1))[s]$, the current value on which $\tau_1$ is depending for the $R^{\tau_1}(k_1)$ substrategy, will be greater than $a_2[s]$. Because of this, the change on $A(a_2)[s]$ which the $\sigma_2$-strategy demands will injure the strategy for keeping

$\Gamma^{\tau_1}_{n_1}(k_1)$ correct. Of course, there are infinitely many other lower-priority strategies similar to that of $\sigma_2$ that may seek to change $A$ and interfere in this way with $\tau_1$, thereby causing infinite injury to the $\tau_1$-strategy from below, a situation which must clearly be avoided. Because of this, we demand that before $\tau_2$ attacks with the $\mathrm{R}^{\tau_2}(k_2)$-attacker, it must first clear the strategy for $\mathrm{R}^{\tau_1}(k_1)$, that is, it must ensure that $\gamma^{\tau_1}(k_1)\uparrow$ at any stage when it acts for the sake of its own requirement. Therefore, before initiating its own attack, $\tau_2$ links up to $\tau_1$, in order to initiate a preliminary attack with the $\mathrm{R}^{\tau_1}(k_1)$-attacker (currently controlled by $\sigma_1$) [3.2, case III.1]. However, this attack may in turn interfere with the still higher-priority $\mathrm{R}^{\tau_0}(k_0)$-strategy, since $a_1[s]$ will in general be less than $\psi_0(A;\phi_0(W^{W_{e_0}}_{l_0};a_0))[s]$. Hence, $\tau_1$ itself must in turn link up to $\tau_0$ [3.2, case III.1], initiating an attack with the $\mathrm{R}^{\tau_0}(k_0)$-attacker [3.2, case III.2]. At the next subsequent $\tau_0$-expansionary stage, the $\mathrm{R}^{\tau_0}(k_0)$-strategy is cleared [3.2, case II.3A]. At this stage, $\tau_1$ requires the restoration of the old value of $\tau_0$'s attacker. Because $\tau_0$ is the highest-priority node, nothing prevents it from restoring the value, so it does so [3.2, case II.3A.2]. Then the $\mathrm{R}^{\tau_1}(k_1)$-strategy proceeds [3.2, case I.1]. At the next subsequent $\tau_1$-expansionary stage, the $\mathrm{R}^{\tau_1}(k_1)$-strategy is cleared [3.2, case II.3]. Now, however, the $\mathrm{R}^{\tau_1}(k_1)$-attacker needs to be restored before the attack for the sake of $\mathrm{R}^{\tau_2}(k_2)$ can proceed. Because this action may interfere with the $\tau_0$-strategy, $\tau_1$ must again link up to $\tau_0$ [3.2, case II.3A.1], which attacks [3.2, case III.1] with the (new) $\mathrm{R}^{\tau_0}(k_0)$-attacker [3.2, case III.2]. At the next $\tau_0$-expansionary stage, $\tau_1$ again requires the restoration of the old value of $\tau_0$'s attacker, and $\tau_0$ does so [3.2, case II.3A.2]. Then the $\mathrm{R}^{\tau_1}(k_1)$-strategy proceeds [3.2, case I.2] to restore its old value for the sake of $\tau_2$, which then is finally able to attack [3.2, case I.1]. At the next subsequent $\tau_2$-expansionary stage, $s'$, the $\tau_2$-attack is completed [3.2, case II.3], with $\gamma^{\tau_2}(k_2)[s']\uparrow$, at which point $\sigma_2$ can achieve its goal of setting $\Gamma^{\tau_2}(k_2) = K(k_2)$. However, $\tau_2$'s work is not yet done, since still lower-priority requirements may be counting on $A(a_2) = A(a_2)[s] \neq A(a_2)[s']$. Because of this, $\tau_2$ must initiate at $s'+1$ what is essentially a repetition of the entire process [3.2, case II.3A.1] in order to restore the original value $A(a_2)[s]$ without injuring $\tau_1$ and $\tau_0$ [a process ending at 3.2, case II.3A.2].

This procedure appears to threaten the totality of $\Gamma^{\tau_1}$, since the value of $\gamma^{\tau_1}(k_1)$ is increased by the (lower-priority) strategy for $\tau_2$. Recall, however, that we demand that $\sigma_2 < \sigma_1$ on the priority tree only if $k_2 < k_1$. Any $\sigma$ with a strategy working for $R^{\tau_2}(k)$ with $k \geq k_1$ will have lower priority and can be forced, therefore, to wait until some stage $t$ where $\psi_1(A;\phi_1(W^{W_{e_1}}_{l_1};a_1))[t]\downarrow$ before picking its attacker $a(k)$. Therefore, only finitely many strategies are in the position of the $\sigma_2$-strategy needing to initiate a sequence of events like the just described. Because only these higher-priority strategies will cause $\gamma^{\tau_1}(k_1)$ to increase, and these will only do so when $K$ changes on the number assigned to them, eventually the strategy for $\sigma_1$ (or some other strategy for $R^{\tau_1}(k_1)$) will be able to pick some $a_1$ permanently, ensuring that $\gamma^{\tau_1}(k_1)\downarrow$.

Of course, things happen differently if clearance is not achieved at some stage, and a master strategy must switch to its backup functional. In this case, an entire process like the one outlined above is cut short, and all strategies believing in the non-backup functional of this master are initialized. The attacker which failed to receive clearance is then made available to the backup strategy. It may already be too small at this stage to actually be used to define a value of the backup functional. For instance, suppose $a$ is a newly available attacker for the backup functional $\Gamma^{\tau}_{\infty}$ for some $\tau$, the next value to be defined is $k$, and $\sigma$ is the substrategy which wishes to set $\Gamma^{\tau}_{\infty}(k)$. Suppose there is some higher-priority $\tau'$ trying to satisfy requirement $\mathrm{R}' = \mathrm{R}_{\Phi',\Psi',e',l'}$, with substrategy $\sigma'$ and $\sigma'$-attacker $a'$ assigned to some $\mathrm{R}'(k')$ such that $\psi'(A;\phi'(W_{l'}^{W_{e'}};a')) > a$. If $\sigma' < \sigma$, then the $\sigma$ strategy cannot be allowed to use $a$ to define $\Gamma^{\tau}_{\infty}(k)$, since this will threaten injury to the higher-priority $\sigma'$-strategy. However, if the attempt by the $\tau$-strategy to build a non-backup functional fails infinitely often, this will generate an infinite stream of available attackers, so that eventually one which is large enough will appear to enable the $\sigma$-strategy to define $\Gamma^{\tau}_{\infty}(k)$.

# 3 The full construction

## 3.1 The priority arrangement

Our notation is standard, as in [5], XIII. We use a priority tree $T$ which is isomorphic to a subtree of ${}^{<\omega}3$. Using standard coding functions for $n$-tuples, as well as standard indexing for computable functionals and computably enumerable sets, we order the requirements in a priority listing. We assign requirements recursively along each path in $T$, achieving this by using two listing functions, $L_1(\beta,k)$ and $L_2(\beta,k)$, which list, for each $\beta \in T$, the requirements that still need to be satisfied at $\beta$. The requirement $L(\beta) = L_1(\beta,0)$ is assigned to $\beta$, if $|\beta|$ is even; and $L(\beta) = L_2(\beta,0)$ is assigned to $\beta$, if $|\beta|$ is odd. A natural notational abbreviation is the writing of $L_j^{\beta}$ for the functional $\lambda x\, L_j(\beta,x)$.

We define $L_1$ and $L_2$ by recursion on $\beta \in T$ and $m \in \omega$, after first making some preliminary definitions.

A node is a *master* if it has even length. A node is a *worker* if it has odd length. Master nodes have outcomes $\infty <_L \mathrm{num} <_L \mathrm{fin}$. Worker nodes have outcomes $\infty <_L \mathrm{fin}$.

We can now define the functions $L_1$ and $L_2$. The intuition is merely that we assign overall requirements in order, and then interleave the subrequirements in one at a time. Below a finite outcome of a master node or an infinite outcome of a worker node, all subrequirements of that strategy are removed from the list $L_2$. Let $\langle\rangle$ be a coding function for pairs such that $\langle m,k\rangle < \langle n,l\rangle$ and $n \le m$ implies $k < l$.

Let $\lambda$ be the empty string.

**Empty string.** For every $m,k \in \omega$, $L_1(\lambda,m) = \mathrm{R}_m$, and $L_2(\lambda,\langle m,k\rangle) = R_m(k)$.

**Master node.** Suppose $\beta$ has requirement $\mathrm{R}_n$ assigned to it. Then $\beta$ has three possible outcomes $\mathcal{O}$.

$\mathcal{O} = \infty$ **or num.** Let $L_1(\beta^\frown\langle\mathcal{O}\rangle, m) = L_1(\beta, m+1)$, and $L_2^{\beta^\frown\langle\mathcal{O}\rangle} = L_2^\beta$.

$\mathcal{O} =$ **fin.** Let $L_1(\beta^\frown\langle\mathcal{O}\rangle, m) = L_1(\beta, m+1)$. Let

$$S(\beta) = \{\, j \,:\, j \neq n \text{ and } \exists i (R_j(i) \in \operatorname{ran}(L_2^\beta))\,\}.$$

Let $f_\beta : \omega \to S(\beta)$ be the enumeration of $S(\beta)$ in increasing order. For every $m, k \in \omega$, $L_2(\beta^\frown\langle\text{fin}\rangle, \langle m, k\rangle) = L_2(\beta, \langle f(m), k\rangle$. (In other words, we just remove $\mathrm{R}_n(k)$ from the range of $L_2$ for every $k$.)

**Worker node.** Suppose $\beta$ has requirement $\mathrm{R}_n(j)$ assigned to it. There are two possible outcomes $\mathcal{O}$.

$\mathcal{O} = \infty$. Let $\beta_0$ be the longest proper substring of $\beta$ with $L(\beta_0) = \mathrm{R}_n$. Let $L_1^{\beta^\frown\langle\infty\rangle} = L_1^\beta$, and let $L_2^{\beta^\frown\langle\infty\rangle} = L_2^{\beta_0^\frown\langle\text{fin}\rangle}$.

$\mathcal{O} =$ **fin**. For every $m \in \omega$, Let $L_1^{\beta^\frown\langle\text{fin}\rangle} = L_1^\beta$, and let, for every $m \in \omega$, $L_2(\beta^\frown\langle\text{fin}\rangle, m) = L_2(\beta, m+1)$.

For any worker node $\beta$ with requirement $\mathrm{R}_n(j)$ assigned to it, the *master of* $\beta$, $\tau(\beta)$, is the greatest $\tau$ included in $\beta$ such that $L(\tau) = \mathrm{R}_n$. We say $\beta$ *must respect* an infinitary outcome of some master node $\tau_0 \subset \beta$ when $\tau^\frown\langle\mathcal{O}\rangle \subseteq \beta$ with $\mathcal{O} = \infty$ or num, and there is no $\sigma_0 \subset \beta$ with $\tau(\sigma_0) = \tau_0$ and $\sigma_0^\frown\langle\infty\rangle \subseteq \beta$. (In other words, when $\beta$ assumes that a $\Pi^0_3$ outcome for $\tau_0$ lies on the true path, and this outcome is not denied by some intermediate node.)

As usual, we have an *approximation to the true path* $f_s$ defined at each $s > 0$. For any node $\beta \in T$, $s$ is a *$\beta$-stage* if $\beta \subset f_s$. If $s$ is an active $\beta$-stage, then we use $s_\beta^-$ to denote the last previous $\beta$-stage. When $\beta$ is clear from the context, we merely write $s^-$ for $s_\beta^-$.

Whenever $f_s <_L \beta$, we *initialize $\beta$ at $s$*. If $\beta$ is a master, this means that we undefine all of $\beta$'s parameters and functionals, and start over completely with a new version of $\beta$. For workers, this means essentially nothing, since the parameters associated to different workers for the same master are the same (see below). At stage 0 we initialize all nodes in $T$. We then take action as follows at each stage $s+1$, breaking the action into substages depending on the order in which the active nodes can act.

## 3.2 Master nodes

Suppose $\tau$ has requirement $\mathrm{R}_{\Phi,\Psi,e,l}$ assigned to it. We first make explicit how we intend to approximate the truth of the condition ($\Phi(W_l^{W_e}) = A$ and $\Psi(A) = W_l^{W_e}$). We use the hat trick.

For each $\tau$-stage $t$ let

$$w_t^\tau = \begin{cases} \mu\, w(w \in W_e[t] - W_e[t^-], & \text{if } W_e[t] - W_e[t^-] \neq \emptyset, \text{ and} \\ t, & \text{otherwise.} \end{cases}$$

Let $\widehat{\Phi_l^\tau}(W_e;x)[t]\downarrow$ if and only if $\phi_l(W_e;x)[t]\downarrow < w_t^\tau$. Let $(\widehat{W_l^{W_e}})^\tau[t] = \{\, x : \widehat{\phi_l^\tau}(W_e;x)[t]\downarrow \,\}$

In other words, $(\widehat{W_l^{W_e}})^\tau[t]$ consists of those elements of $W_l^{W_e}[t]$ with axioms smaller than $w_t^\tau$. A stage $t$ is said to be a *$\tau$-true stage*, if $t$ is a $\tau$-stage and $W_e \restriction w_t^\tau = W_e[t] \restriction w_t^\tau$. This means that no element $w < w_t^\tau$ is ever enumerated into $W_e$ at any stage after $\tau$.

Let $s$ be a $\tau$-stage. We define the set $S^\tau[s]$ of *apparent $\tau$-true stages at $s$* to be the set of $\tau$-stages $t < s$ such that for all $t' \leq s$, if $t < t'$ and $t'$ is an active $\tau$-stage, then $w_t^\tau < w_{t'}^\tau$. When a fixed $\tau$ is under consideration, we usually write $w_t$ for $w_t^\tau$ and $\widehat{W_l^{W_e}}$ for $(\widehat{W_l^{W_e}})^\tau$, and we call $\tau$-true stages $W_e$-true stages.

At each $\tau$-stage $t$, we define the *$\tau$-length-of-agreement at $t$*, $l^\tau[t]$, to be the greatest $x$ such that for every $y < x$, $\Phi(\widehat{W_l^{W_e}};y)[t] = A(y)[t]$ and for every $z < \phi(\widehat{W_l^{W_e}};y)[t]$, $\widehat{W_l^{W_e}}(z)[t] = \Psi(A;z)[t]$. We define the *maximum previous $\tau$-length-of-agreement at $t$* by $m^\tau[t] = \max\{\, l^\tau[s] : s \text{ a } \tau\text{-stage and } s < t \,\}$. A $\tau$-stage $t$ is *$\tau$-expansionary* whenever $l^\tau[t] > m^\tau[t]$.

We remind the reader of the main features of the hat trick. The significance of true stages lies in the following fact: If there exist infinitely many $\tau$-stages and $u$ is any natural number, then there exists a least $\tau$-true stage $t(u)$ such that for all $t \geq t(u)$, if $t$ is a $\tau$-true stage, then $\widehat{W_l^{W_e}}[t] \restriction u = W_l^{W_e} \restriction u$. Suppose there are infinitely many $\tau$-stages, $\Phi(W_l^{W_e}) = A$, and $\Psi(A) = W_l^{W_e}$. Then, if $A$ is a $\Delta_2^0$ set, every relevant computation eventually appears cofinitely often in the sequence of $\tau$-true stages. In this case, there will exist infinitely many $\tau$-expansionary stages. This means our approximation will be good enough for us to satisfy $\mathrm{R}_{\Phi,\Psi,e,l}$. (To allay any fears that our argument may be circular, we remark here that the proof that $A$ is $\Delta_2^0$, in fact, 3-c.e., will be independent of the existence of infinitely many $\tau$-expansionary stages.)

Recall that when some substrategy of $\tau$ is successful, we need to go through a procedure to restore the state of $A$ before this strategy acted. This is how lower-priority requirements avoid being injured infinitely often. This gives rise to two distinct states $\tau$ can be in, depending on whether it is aiming at permission for an initial attack, or for restoration of an old value. Below, we divide $\tau$'s action during an attack into two parts. The first part begins when some lower priority nodes links up $\tau$ because it wishes $\tau$ to make some initial attack. After the first $\tau$-action to change $A$, the second part of the attack begins. This is to signal that at the next $\tau$-expansionary stage $\tau$ must attempt to change $A$ back to its former state, rather than following the link back down from $\tau$, because the node that was waiting for the original $\tau$-attack to succeed, will in general (*i.e.*, when it is a lower-priority master) require restoration of this old value. At this point, $\tau$ itself may have to wait a while for permission from higher-priority masters to restore the value, but eventually it does so, and then, at the next $\tau$-expansionary stage, we consider $\tau$'s attack completed, we can follow the link down, allowing the lower-priority node to proceed. It may help the reader's intuition in understanding what follows for the reader to note explicitly that initial attacks occur under cases I.1 and III.2 below, while restoration occurs under cases I.2 and II.3A.2.

A possible source of confusion is the suppression of any indexing of the successive attempts to define $K \leq_T W_e$ without recourse to the backup functional. This involves constructing some $\Gamma^\tau_n$ where $n$ is the current attempt at computing $K$ below the 'num' outcome. This $n$ is fixed in the intervals throughout which it appears to be succeeding, and is incremented by one every time there is an uncorrectable failure, at which point it is given up forever. There is no need to make any mention of this $n$: in fact, this would do nothing but add notational complexity to what follows. For this reason, the current $\Gamma^\tau_n$ appears as $\Gamma^\tau$ below. We write $a(\tau, k)[s]$ for the current $k$th attacker for $\tau$'s non-backup functional, and $a_\infty(\tau, k)[s]$ for the backup functional's $k$th attacker. In order to set appropriate restraints on $A$, we also keep track of the stage at which these attackers become defined with their current values, by means of parameters $s^\tau(k)[s]$, and $s^\tau_\infty(k)[s]$, respectively.

Recall the description of the general plan for satisfying $\tau$'s requirement in section 2. The backup functional built by the $\tau$-strategy will be total only if the attempt to build a non-backup functional fails infinitely often. If this happens, the infinite sequence of numbers on which these failures have occurred can be used as attackers in defining values of the backup functional which are guaranteed to be correct. As described at the end of section 2.2, the substrategies defining $\gamma^\tau_\infty$ must choose numbers large enough to avoid injury to higher priority requirements, and hence only a subsequence of this sequence of numbers can actually be used. We control the sequence of numbers on which failures have already occurred at stage $s$ with an availability list $\mathcal{A}^\tau_\infty$. These are the numbers which are available to substrategies working to define $\Gamma^\tau_\infty$ from which they must choose large enough numbers as their attackers. This "streaming" of available numbers is somewhat different from that of Downey [2], since only substrategies of this overall strategy have to select from the stream. We let $\mathcal{A}^\tau_\infty[s] = \emptyset$ when $s = 0$, and at any stage $s+1$ at which either $\tau$ is initialized or a new attacker is selected from $A^\tau_\infty[s]$; and we gradually add numbers to $\mathcal{A}^\tau_\infty$ as more and more failures occur. (See case II.3B below.)

There are three different situations in which $\tau$ can be allowed to act at stage $s$. $\tau$ can either be visited by a link from some master node $\tau_0 \subset \tau$; or $\tau$ can be visited in the ordinary way, by being the single outcome extension of some $\mu$ which acted at $s$; or, finally, some link with top $\tau$ can be set by some $\rho$ with $\tau \subset \rho$ for the purpose of initiating a $\tau$-attack. In the final case, it may be that $\rho$ is itself a master node working for a different requirement which is trying to clear some $\tau$-substrategy $\mathrm{R}^\tau(k)$ in order to get permission to act for one of its strategies $\mathrm{R}^\rho(l)$. In this case we say that the $\mathrm{R}^\tau(k)$-substrategy is *associated to the link* which is being set at this stage, and we say that the $\mathrm{R}^\rho(l)$-strategy is *waiting for the* $\mathrm{R}^\tau(k)$ *strategy to be cleared.* To facilitate our description of the action we make a formal definition of when some master needs to obtain clearance from a higher priority master in order to act.

**Definition 1.** Suppose $\tau$ is a master node, $k \in \omega$, and either $s^\tau(k)[s]\downarrow$, or

$s^\tau_\infty(k)[s]\downarrow$. Let $s(k)$ denote either of these, and let $a[s]$ be the attacker associated to $s(k)$ (i.e., either $a(\tau,k)[s]$ or $a_\infty(\tau,k)[s]$). Suppose there is another master node $\tau_0$ such that either

- $\tau_0^\frown\langle\text{num}\rangle \subseteq \tau$ and there is a least $k_0$ such that $(s(k) < s^{\tau_0}(k_0))[s]$, or
- there is a node $\tau_0$ such that $\tau_0^\frown\langle\infty\rangle \subseteq \tau$ and there is a least $k_0$ such that $(s(k) < s^{\tau_0}_\infty(k_0))[s]$.

Then we say $\tau$ *requires clearance from* $\tau_0$ *before changing* $A$ *on* $a[s]$.

After making these preliminary remarks, we can finally give the possibilities for the action of $\tau$. At stage 0, all nodes are initialized by undefining all functions involved in their strategies and setting all sets equal to $\emptyset$. There are three sets of possibilities at stage $s+1$, depending on how $\tau$ is visited at stage $s+1$. For each of these situations, the first possibility below that applies is the one that is followed.

**I.** Suppose $\tau$ is visited by a link from some other master node $\tau_0 \subset \tau$ (acting at the immediately preceding substage). Such a link is originally set under one of cases II.3A.1 or III.1 below when some $\tau$-strategy wished to change $A$ but was prevented from doing so because of the injury this would have caused to some $\mathrm{R}^{\tau_0}(k_0)$-substrategy which has now been cleared. Therefore, the $\tau$-strategy has just received permission to act. There are two subcases for action, depending on which part of the current $\tau$-attack is under way. (Note that both parts of $\tau_0$'s attack must have been completed, otherwise $\tau$ could not be visited by a link, by Case I.1 applied to $\tau_0$.)

**Case I.1** Suppose $\tau$ is in part one of its current attack, and there is a link with top $\tau$ and bottom $\rho$ in place. If $\rho$ is not a worker for $\tau$, then such a link can only be set under case III.1 below, and there will in this case be an associated $\mathrm{R}^\tau(k)$-substrategy, for some $k \in \omega$. Otherwise, the link was set under case II.3A.1 below, and $\rho$ is a worker for $\tau$ with requirement $\mathrm{R}_n(k)$, for some $k \in \omega$. If $\tau^\frown\langle\text{num}\rangle \subseteq \rho$, let $A(a(\tau,k))[s+1] = 1 \neq A(a(\tau,k))[s^\tau(k)[s]]$. If $\tau^\frown\langle\infty\rangle \subseteq \rho$, let $A(a(\tau,k))[s+1] = 0 \neq A(a(\tau,k))[s^\tau_\infty(k)[s]]$. Immediately end stage $s+1$ and proceed to stage $s+2$. (At the next $\tau$-expansionary stage, $\tau$ will act under case II.3 below.)

**Case I.2** Suppose $\tau$ is in part two of its current attack, and there is a link with top $\tau$ and bottom $\rho$ in place. If $\rho$ is not a worker for $\tau$, then, as in I.1, there will again be an associated $\mathrm{R}^\tau(k)$-substrategy, for some $k \in \omega$. (In this case $\rho$ is a lower priority master that needed clearance from $\tau$, as in the case of $\tau_1$ in the detailed example of section 2.2 above.) Otherwise, $\rho$ is a worker for $\tau$ with requirement $\mathrm{R}_n(k)$, for some $k \in \omega$. We say $\tau$ *has completed both parts of its current attack*. If $\tau^\frown\langle\text{num}\rangle \subseteq \rho$, let $A(a(\tau,k))[s+1] = 0$, and let

$a(\tau, k)[s+1]\uparrow$. If $\tau^\frown\langle\infty\rangle \subseteq \rho$, let $A(a_\infty(\tau, k))[s+1] = 1$, and let $a_\infty(\tau, k)[s+1]\uparrow$. Remove the link, and allow $\rho$ to act at stage $s+1$.

**II.** Suppose $\tau$ is visited in the ordinary way at stage $s$, because $\tau = \lambda$, or $\tau = \mu^\frown\langle\mathcal{O}\rangle$, for some $\mu$ which acted at stage $s$ and received outcome $\mathcal{O}$.

**Case II.1.** Suppose $s$ is not $\tau$-expansionary. Let $\tau^\frown\langle\text{fin}\rangle$ act at stage $s+1$.

**Case II.2.** Suppose $s$ is $\tau$-expansionary and there is no link with top $\tau$ in place. (This means we continue in the belief that for the current $n$, $\Gamma_n^\tau = K$.) Let $\tau^\frown\langle\text{num}\rangle$ act at stage $s+1$.

**Case II.3.** Suppose $s$ is $\tau$-expansionary, there is a link with top $\tau$ and bottom $\rho$ in place, and $\tau$ is in part one of some current attack. Because $\tau$ is the top of a link, there exists a $k$ such that either $\rho$ is a worker for $\tau$ with requirement $\mathrm{R}_n(k)$; or $\rho$ is not a worker for $\tau$ and there is some associated $\mathrm{R}^\tau(k)$-substrategy.

There are two possible subcases, depending on whether this part of the attack has been successful or not.

**Case II.3A.** Suppose $\tau^\frown\langle\infty\rangle \subseteq \rho$, or $\tau^\frown\langle\text{num}\rangle \subseteq \rho$ and $\gamma^\tau(k)[s]\uparrow$. This means the substrategy for $\mathrm{R}^\tau(k)$ has been cleared so that $\rho$ may proceed without injuring the $\tau$-strategy; however $\tau$ must now restore the state of $A$ which $\rho$ may have been depending on when the $\tau$-attack was started. In this case there are two further subcases depending on whether $\tau$ requires permission before restoring the previous state of $A$. Let $a[s]$ be either $a(\tau, k)[s]$ or $a_\infty(\tau, k)[s]$, depending on which outcome is included in $\rho$.

**Case II.3A.1.** Suppose there exists some node $\tau_0$ such that $\tau$ requires clearance from $\tau_0$ before changing $A$ on $a[s]$. (Here $\tau$ is in the position of $\tau_2$ and $\tau_1$ in the example of section 2.2.) Let $\tau_0$ be the longest (*i.e.*, lowest priority) such node. Set a link between $\tau$ and $\tau_0$, and declare the $R^{\tau_0}(k_0)$ strategy temporarily associated to the link between $\tau$ and $\tau_0$. We say $\tau$ *enters part two of its current attack.* Allow $\tau_0$ to take appropriate action (under case III.1 or III.2 below) at stage $s+1$. (The $\mathrm{R}^\tau(k)$ strategy is now waiting for the $\mathrm{R}^{\tau_0}(k_0)$ strategy to be cleared.)

**Case II.3A.2.** Otherwise, $\tau$ may immediately restore its previous value and allow $\rho$ to proceed. If $\tau^\frown\langle\text{num}\rangle \subseteq \rho$, then let $A(a)[s+1] = 0$, and let $a[s+1]\uparrow$. If $\tau^\frown\langle\infty\rangle \subseteq \rho$, let $A(a)[s+1] = 1$, and let $a[s+1]\uparrow$. We say $\tau$ *has completed both parts of its current attack.* Remove the link, and allow $\rho$ to act at stage $s+1$.

**Case II.3B.** Suppose $\tau^\frown\langle\text{num}\rangle \subseteq \rho$ and $\gamma^\tau(k)[s]\downarrow$. (This means the substrategy for $\mathrm{R}^\tau(k)$ has failed.) Declare $a(\tau, k)[s]$ to be *available below* $\tau^\frown\langle\infty\rangle$, and let $a(\tau,k)[s+1]\uparrow$. We say $\tau$ has completed both parts of its current attack (through failure), and let the entire functional $\Gamma^\tau$ be undefined. Let $\tau^\frown\langle\infty\rangle$ act at stage $s+1$. (In this case, $\rho$ is initialized.)

**III.** Suppose a link is set at stage $s$ with top $\tau$ and bottom either some $\rho$ with requirement $\mathrm{R}^\tau(k)$, or some $\rho$ which is not a worker for $\tau$. In the latter case there is some associated $\mathrm{R}^\tau(k)$-substrategy. We write $s(k)$ for either $s^\tau(k)$ or $s^\tau_\infty(k)$, depending on which of $\tau^\frown\langle\mathrm{num}\rangle$ and $\tau^\frown\langle\infty\rangle$ are included in $\rho$. This situation arises when we wish to change some value of $A$ for the sake of the $\tau$-strategy. As in II.3A above, there are two possibilities, depending on whether this change in $A$ threatens to injure some higher priority strategy (III.1), or not (III.2). In either case, we say $\tau$ *enters part one of its current attack*. Let $a[s]$ be either $a(\tau,k)[s]$ or $a_\infty(\tau,k)[s]$, depending on which outcome is included in $\rho$.

**Case III.1** Suppose there is a node $\tau_0$ such that *tau* Suppose there exists some node $\tau_0$ such that $\tau$ requires clearance from $\tau_0$ before changing $A$ on $a[s]$. Let $\tau_0$ be the longest (*i.e.*, lowest priority) such node. Set a link between $\tau$ and $\tau_0$, and declare the $R^{\tau_0}(k_0)$ strategy temporarily associated to the link between $\tau$ and $\tau_0$. Allow $\tau_0$ to take appropriate action (under cases III.1 or III.2) at stage $s+1$. (The $\mathrm{R}^\tau(k)$ strategy is now waiting for the $\mathrm{R}^{\tau_0}(k_0)$ strategy to be cleared.)

**Case III.2** Otherwise, $\tau$ may immediately begin its current attack. If $\tau^\frown\langle\mathrm{num}\rangle\subseteq\rho$, then let $A(a)[s+1]=1\neq A(a)[s^\tau(k)[s]]$. If $\tau^\frown\langle\infty\rangle\subseteq\rho$, let $A(a)[s+1]=0\neq A(a)[s^\tau_\infty(k)[s]]$. Immediately end stage $s+1$ and proceed to stage $s+2$. (At the next $\tau$-expansionary stage, $\tau$ will act under case II.3.)

## 3.3 Worker nodes

Worker nodes are those which have the responsibility of defining and keeping correct the individual values of the functionals which compute $K$. Suppose $\sigma$ is such a node with subrequirement $\mathrm{R}_n(k)=\mathrm{R}_{\Phi,\Psi,e,l}(k)$ assigned to it.

Recall from section 3.2 that the master of $\sigma$, $\tau(\sigma)$, is the longest $\tau$ included in $\sigma$ such that $L(\tau)=\mathrm{R}_n$.

There are two sets of possibilities, depending on whether or not $\sigma$ is a subrequirement for building the backup functional $\Gamma^{\mathrm{R}_n}_\infty$. The procedures for these two different kinds of workers are almost identical, differing only in one case, 4, below for which we distinguish a prime and a non-prime version. Thus, we abuse notation slightly and write $\gamma^\tau(k)$ for both $\gamma^\tau(k)$ and $\gamma^\tau_\infty(k)$ in all cases except 4, and we do the same for $a(\tau,k)$. Although $\sigma$ only has the responsibility to set up and keep correct a single value of some $\Gamma^{\mathrm{R}_n}$, its action is complicated a little by its need to wait until higher priority workers have succeeded in setting up their own strategies. At stage $s+1$, we act according to the first case which applies below. Recall that $s^-$ is the last previous $\sigma$-stage.

**Case 1.** Suppose $\sigma$ has previously been visited as the bottom of a link since it was last initialized. Then $\sigma$'s strategy has finished, and we do not wish it to interfere with any other strategy below. Let $\sigma^\frown\langle \text{fin} \rangle$ act at stage $s+1$.

**Case 2.** Suppose $k \in K[s+1]$ and either

- $\gamma^\tau(k)[s]\uparrow$, or
- $\gamma^\tau(k)[s]\downarrow = \gamma^\tau(k)[s^-]$, and $\Gamma^\tau(k)[s] = 1$.

In this case, $\sigma$'s strategy has succeeded, in the first case, possibly by $\sigma$ being visited as the bottom of a link. If $\gamma^\tau(k)[s]\uparrow$, then, if $\gamma^\tau(k)[s^-]\downarrow$, let $\gamma^\tau(k)[s+1]\downarrow = \gamma^\tau(k)[s^-]$; otherwise (if $\gamma^\tau(k)[s^-]\uparrow$), let $\gamma^\tau(k)[s+1]\downarrow = s+1$. In either case, let $\Gamma^\tau(k)[s+1] = 1$. If $a(\tau,k)\uparrow[s]$, set $a(\tau,k)[s+1] = 0$. If $\gamma^\tau(k)[s]\downarrow = \gamma^\tau(k)[s^-]$, and $\Gamma^\tau(k)[s] = 1$, do nothing. If $\sigma$ was visited as the bottom of a link at stage $s+1$, immediately end stage $s+1$ and go to stage $s+2$. Otherwise, let $\sigma^\frown\langle \text{fin} \rangle$ act at stage $s+1$.

**Case 3.** Suppose there is a master node $\tau_0$ such that either

- $\tau_0^\frown\langle \text{num} \rangle \subseteq \tau(\sigma)$, $\sigma$ must respect this infinitary outcome of $\tau_0$, and there is a $k_0 < k$ such that $a(\tau_0, k_0)[s]\uparrow$ or $(l^{\tau_0} < a(\tau_0, k_0))[s]$; or
- $\tau_0^\frown\langle \infty \rangle \subseteq \tau(\sigma)$, $\sigma$ must respect this infinitary outcome of $\tau_0$, and there is a $k_0 < k$ such that $a_\infty(\tau_0, k_0)\uparrow$ or $(l^{\tau_0} < a_\infty(\tau_0, k_0))[s]$.

In this case, $\sigma$ must wait for a higher priority attack to be prepared. End stage $s+1$, and go immediately to stage $s+2$.

**Case 4.** Suppose $a(\tau,k)[s]\uparrow$ and $\tau^\frown\langle \text{num} \rangle \subseteq \sigma$. Since case 3 does not hold, $\sigma$ can start the $\mathrm{R}_n(k)$-strategy. Let $a(\tau,k)[s+1]$ be the least number greater than any yet mentioned in the construction. Immediately end stage $s+1$, and go to stage $s+2$.

**Case 4′.** Suppose $a_\infty(\tau,k)[s]\uparrow$ and $\tau^\frown\langle \infty \rangle \subseteq \sigma$. Since case 3 does not hold, $\sigma$ can start the $\mathrm{R}_n(k)$-strategy; here, however, we must take extra steps to ensure that the attacker chosen is big enough, since arbitrarily large numbers are not available to $\sigma$. Let $T(\sigma)[s]$ be the set of all master nodes with infinitary outcomes included in $\sigma$ which $\sigma$ must respect. Let $r(\sigma)[s] = \{\, s^\rho(k_0)[s+1] \,:\, k_0 \leq k \text{ and } \rho \in T(\sigma)[s] \,\}$. If there is an available attacker $a$ below $\tau^\frown\langle \infty \rangle$ such that $a > r(\sigma)[s]$, then choose the least such $a$, let $a_\infty(\tau,k)[s+1] = a$, declare $a$ no longer available, and reset $\mathcal{A}^\tau_\infty = \emptyset$. Otherwise, do nothing. In either case, immediately end stage $s+1$, and go to stage $s+2$.

**Case 5.** Suppose $a(\tau,k)[s]\downarrow$ and $(l^\tau \leq a(\tau,k))[s]$. $\sigma$ must then continue to wait for its strategy to be prepared. Immediately end stage $s+1$, and go to stage $s+2$.

**Case 6.** Suppose $a(\tau,k)[s]\downarrow$ and $(l^\tau > a(\tau,k))[s]$, $k \notin K[s+1]$, and $\gamma^\tau(k)[s]\uparrow$. Now $\sigma$ can set the use $\gamma^\tau(k)$. Let $\gamma^\tau(k)[s+1] = \max(\{\, \phi_l(y)[s] : y \in W_l^{W_e} \upharpoonright \phi(x)[s] \,\})$, $\Gamma^\tau(k)[s+1] = 0$, and $s^\tau(k)[s+1] = s$. Let $\sigma^\frown\langle\infty\rangle$ act at stage $s+1$.

**Case 7.** Suppose $a(\tau,k)[s]\downarrow$, $\gamma^\tau(k)[s]\downarrow$ and $\Gamma^\tau(k)[s+1] = K(k)[s+1]$. Let $\sigma^\frown\langle\text{fin}\rangle$ act at stage $s+1$.

**Case 8.** Suppose $a(\tau,k)[s]\downarrow$, $\gamma^\tau(k)[s]\downarrow$, and $\Gamma^\tau(k)[s+1] \neq K(k)[s+1]$. In this case, $\sigma$ *initiates an attack*. We set a link between $\sigma$ and $\tau$ and allow $\tau$ to take appropriate action (under cases III.1 or III.2 of section 3.2) at stage $s+1$.

This completes the construction.

# 4 Verification

We must show that $A$ is 3-c.e. and that every requirement $\mathrm{R}_{\Phi,\Psi,e,l}$ is satisfied. In what follows, we assume familiarity with $\mathbf{0}'''$-priority constructions, to avoid having to prove some tedious technical facts, for example that all requirements that need to be satisfied are eventually assigned to some node along the true path.

**Lemma 4.1.** *$A$ is 3-c.e.*

*Proof.* Suppose $a \in \omega$ is eventually chosen as an attacker for some substrategy of the construction. The value $A(a)[s]$ can only change under cases I, II.3 or III.2 of section 3.2. If this change occurs under cases I.2 or II.3, it results in the permanent abandonment of $a$ as an attacker in the construction. An examination of these cases shows that an initial change on $A(a)[s]$ can only happen below a num outcome of a master node in the first part of an attack and, hence, must occur under cases I.1 or III.2. The only way in which an original change under one of these cases can fail to be followed by restoration and abandonment of the attacker $a$ is under case II.3B, since it is not hard to see that the use tied to $a$ (*i.e.*, some $\gamma^\tau(k)$) must be undefined when $\tau$ enters the second part of its attack. Neither of these situations causes a change in $A(a)[s]$, and each of them reserves $a$ permanently for use as an attacker for the sake of a backup strategy. If $A(a)$ changes for a second time, this must again occur under case I.1 or III.2 for the sake of some substrategy below an $\infty$. In this case, however, the change must be followed by subsequent change under I.2 or II.3 when the next link is removed, which is final as noted. This means that at most three changes of value are possible. □

We now show, using a sequence of lemmas, that each requirement is satisfied. We define the *true path* to be $f = \liminf_s f_s$. We first show that nodes on the true path are not linked over infinitely often.

**Lemma 4.2.** *If $\rho \subset f$, then $\forall s \exists t > s\,(\rho \subset f_t$ and $\rho$ acts at $t)$.*

*Proof.* Suppose not and choose $\rho$ of shortest length such that the lemma fails for $\rho$. Let $\rho^-$ be $\rho$'s immediate predecessor on $f$, so that $\rho^-$ acts infinitely often. We assume all action takes place after $\rho$ is right of the approximation to $f$ for the last time. Given a stage $s$, let $t_0 > s$ be a stage at which $\rho^-$ acts and $\rho \subset f_{t_0}$. If $\rho$ does not act at stage $t_0$, $\rho^-$ must already be linked over $\rho$ at stage $t_0 - 1$ and this link must be removed at stage $t_0$. Because $\rho^-$ may be the top of a chain of links, rather than just a single link to a worker, the bottom of this link may be a master node acting under 3.2, case I.2, at stage $t_0$. However, any chain of links must end in some worker node; hence, the node of greatest length below $\rho^-$ which acts at stage $t_0$ must act under 3.2, case I.1, or 3.3, case 2. Links can only be set from below, and no new link to a node extending $\rho^-$ is set at stage $t_0$, since all these nodes act under either 3.2, case I, or 3.3, case 2 at stage $t_0$. But then $\rho$ itself must act at the next stage $t$ such that $\rho \subset f_t$. $\square$

If $\rho \subset f_s$ and $\rho$ is not linked over at $s$, then we call $s$ an *active $\rho$-stage.* We first prove a lemma which will eventually enable us to show that our procedure succeeds in defining total functions. This will also enable us to show that $f$ is infinite. The latter is not immediately obvious, since $f_s$ fails to be extended when some worker is waiting under 3.3, case 3 for the appearance of an attacker for some higher priority master with an infinitary outcome, and for the associated length-of-agreement function to increase beyond this number. ($f_s$ can also fail to be extended when it is visited as the bottom of a link in case 2, and under cases 4, 5, and 8, but each of these is a trivial case for the induction, since it can only happen once for each node after initialization.) We only have to consider the case where $k \notin K$, since otherwise eventually, for any master $\tau$, any use $\gamma^\tau(k)$ and attacker $a(\tau, k)$ are continually reset to the same number, by 3.3, case 2, and hence must converge. For $\gamma^\tau(k)$ this follows because $W_e$ is c.e.

**Lemma 4.3.** *Let $\tau \subset f$ with requirement* R *assigned to $\tau$. Suppose $k \notin K$, let $\sigma \subset f$ be a worker for $\tau$ with $L(\sigma) = \mathrm{R}(k)$, and suppose there are infinitely many active $\sigma$-stages.*

*If $\tau^\frown\langle num\rangle \subset f$, then for almost all $s$, $a(\tau, k)[s]\downarrow$.*

*If $\tau^\frown\langle\infty\rangle \subset f$, then for almost all $s$, $a_\infty(\tau, k)[s]\downarrow$.*

*Proof.* Suppose otherwise, and choose $\sigma$ of least length for which this fails, and as in the statement of the lemma, let $\tau$ be $\sigma$'s master. As pointed out above, $\sigma$ must be the only node on $f$ which fails to have an outcome on a cofinite sequence of stages. Let $a[s] = a(\tau, k)[s]$ or $a_\infty(\tau, k)[s]$, respectively, depending on whether $\tau^\frown\langle\mathrm{num}\rangle \subset f$ or $\tau^\frown\langle\infty\rangle \subset f$. Let $\mathcal{O}$ be the (infinitary) outcome of $\tau$ on $f$. Let $s_0$ be a stage such that for all $s \geq s_0$, $\sigma \leq f_s$. We first show that $a[s]\downarrow$ at infinitely many $s$, then that it is defined cofinitely often. First, suppose $\mathcal{O} = \infty$. By lemma 4.2, no $\tau_0 \subset \tau$ can link over $\tau$ cofinitely often. Because of this, if the sequence of numbers available below $\tau^\frown\langle\infty\rangle$ were bounded, then, after the link to the lowest priority node working for $\tau$ with a defined attacker below $\tau^\frown\langle\infty\rangle$ is removed, no further link could be imposed with top $\tau$ and bottom extending $\tau^\frown\langle\infty\rangle$ without $\tau^\frown\langle\infty\rangle$ acting. Hence, $\tau$ itself could not be the top of a link at cofinitely-many $\tau^\frown\langle\infty\rangle$ stages. There must then be infinitely many stages at which $\tau^\frown\langle\infty\rangle$ acts. But at any such stage, $\tau$ has acted under case II.3B, and so a new number has been made available below $\tau^\frown\langle\infty\rangle$. Also, if $\tau' \subset \sigma$ and $k' \in \omega$, then $s^{\tau'}(k')$ is only set under 3.3, case 7 for some $\sigma'$, at which point $\sigma'^\frown\langle\infty\rangle$ acts. No node extending $\sigma'^\frown\langle\infty\rangle$ respects $\tau'$. By inductive hypothesis, all workers $\sigma' \subset \sigma$ with masters that $\tau$ must respect eventually define their attackers permanently. This also means that the restraint defined in 3.3, case $4'$ on $\sigma$ is bounded. Hence, $a_\infty(\tau,k)[s]\downarrow$ infinitely often.

If $\mathcal{O} = \text{num}$, then $a(\tau,k)[s]$ can always be chosen under 3.3, case 4. Again, by hypothesis, $\sigma$ cannot be kept waiting under case 3 forever. So, no matter what $\mathcal{O}$ is, $a[s]$ converges infinitely often.

We may assume that no number less than $k$ enters $K$ at any stage after $s_0$, by choosing $s_0$ larger, if necessary. Since $k \notin K$, $a[s]$ can never be given up as a result of an attack for the sake of the $\tau$-strategy. At every active $\sigma$-stage all masters to the right of the true path are initialized, hence $a[s]$ can never diverge for the sake of such a node. This means that it is only a substrategy of some master node $\rho$ such that $\tau \subset \rho \subset \sigma$ acting under 3.2 , case III.1 that can cause $a[s]$ to become undefined infinitely often.

Let $\rho$ be the longest such node included in $\sigma$ and let $t_0$ be the stage at which the R$(k)$ strategy was temporarily associated to a link between $\tau$ and $\rho$. We may assume that all the nodes $\mu \subset \sigma$ which cause $a[s]$ to become undefined only finitely often do so only at stages before $t_0$. Recall that $K \restriction k = K[t_0] \restriction k$. Let $R^\rho(n)$ be the requirement whose strategy is waiting for the R$(k)$ strategy to be cleared. By section 3.3, case 3, $n < k$. Now, if the $\mathrm{R}^\rho(n)$ strategy were itself acting at $t_0$ because it had in turn been associated temporarily to some link between $\rho$ and some lower-priority master node $\mu$, then, $\mu > \sigma$, since otherwise $\rho$ is not the longest node included in $\sigma$ which affects $a[s]$ at any stage greater than or equal to $t_0$. But then, for any $m \in \omega$, $s^\mu(m)$ and $s^\mu_\infty(m)$ (if defined at all) are both greater than whichever of $s^\tau(k)$ and $s^\tau_\infty(k)$ is defined for $a[t_0]$. But then they are *a fortiori* greater than whichever of $s^\rho(n)$ and $s^\rho_\infty(n)$ causes the attack with $a[t_0]$ to happen at $t_0$. (In other words, using an obvious but sloppy notation, $s^\rho(m) < s^\tau(k) < s^\mu(m)$.) But this means such a $\mu$ cannot set a link to $\rho$ because of any substrategy. This implies that the strategy for $R^\rho(n)$ is acting on its own behalf, so that $n \in K[t_0]$. Without loss of generality, we can assume $n$ is the least number that causes this kind of activity to occur for the overall $R^\rho$ strategy. But then, after stage $t_0$, since $n < k$, we must have all $m \in K$ such that $n < m < k$ elements of $K[t_0]$. Hence no more $R^\rho$ strategies for any number less than $k$ can be subsequently started until after $a[s]\downarrow$ at some $s > t_0$. Thus $\rho$ can never again affect $a[s]$. This is a contradiction. □

As pointed out above, lemma 4.3 implies that the true path is infinite, since only under 3.3, case 3 can a node fail to have an outcome at many stages. It follows straightforwardly from the definitions in 3.1 that every requirement $\mathrm{R} = \mathrm{R}_{\Phi,\Psi,e,l}$ is assigned to some greatest node along every infinite path in $T$. In what follows, we let $\tau$ be the unique such node on $f$. We assume that for every master $\tau_0 \subset \tau$, $\tau_0$'s requirement is satisfied, and if $\tau_0$ has an infinitary outcome, then the functional associated to that outcome is totally defined and correct.

As discussed in section 3.2, the fact that $A$ is $\Delta^0_2$ implies that the $\tau$-length-of-agreement function increases infinitely often if the $\tau$-condition, ($\Phi(W_l^{W_e}) = A$ and $\Psi(A) = W_l^{W_e}$), is satisfied. Hence, if $\tau^\frown\langle\text{fin}\rangle \subset f$, the requirement is satisfied. Also, if some $\sigma$ is a worker for $\tau$ and $\sigma^\frown\langle\infty\rangle \subset f$, then the total use involved in $\Phi(W_l^{W_e}; a) = A(a)$ and $\Psi(A) \upharpoonright \phi(a) = W_l^{W_e} \upharpoonright \phi(a)$ must increase without bound on expansionary stages. (Here $a$ is the final value for the $\sigma$-strategy's parameter.) Again, since $A$ is $\Delta^0_2$, this cannot happen if the $\tau$-condition is satisfied. Hence we only have to consider the situation where $\tau$ has an infinitary outcome on the true path and every worker for $\tau$ on the true path has a finitary outcome. In what follows, we assume that this condition is satisfied, and that all our discussion takes place after the last stage at which the approximation to the true path branches back left of $\tau$.

Because every master $\tau_0 \subset \tau$ is able to define its functional correctly, $\tau$'s immediate predecessor must have a true outcome infinitely often. This implies that there are infinitely many active $\tau$-stages. To show that our linking procedure works correctly, however, we need to show that workers for $\tau$ along the true path also receive infinitely many chances to act.

We next prove the technical fact which implies that higher-priority strategies either succeed in restoring $A$ to the state which lower priority strategies expect, or initialize those strategies completely.

**Lemma 4.4.** *Let $\rho_0 \subset \rho_1$. Suppose a link is set at stage $s_0$ between $\rho_0$ and $\rho_1$. Let $a$ be the attacker on which $\rho_0$ wishes to change $A$'s value at stage $s_0$, and let $s_1$ be the next active $\rho_1$ stage, if such a stage exists. Then either*

*i.* $(A \upharpoonright s_0)[s_1] = (A \upharpoonright s_0)[s_0]$, *or*

*ii.* $\rho_0$ *has been initialized at some stage $t$ such that* $s_0 < t < s_1$, *or*

*iii.* $\rho_0^\frown\langle num\rangle \subseteq \rho_1$, $\rho_0^\frown\langle num\rangle$ *has been initialized at some stage $t$ such that* $s_0 < t < s_1$, *and* $(A \upharpoonright s_0)[s_1] = (A \upharpoonright s_0)[s_0] \cup \{a\}$.

*Proof.* By induction. Suppose this fails for some shortest $\rho_0 \subset \rho_1$. No node to the left of $\rho_0$ can act again before stage $s_1$ (since *ii* fails), every node to the right of $\rho_0$ picks witnesses bigger than $s_0$, and every node between $\rho_0$ and $\rho_1$ is prevented from acting while the link is in place. So, since the claim never failed before, whenever $\rho_0$ acts, it can depend on $A$ having the right state except for the attacker $a$; otherwise $\rho_0$ is initialized by some even higher-priority strategy before stage $s_1$. By the failure of $i$, we must therefore have one of two possibilities: either $(A \upharpoonright s_0)[s_1] = (A \upharpoonright s_0)[s_0] \cup \{a\}$, or $(A \upharpoonright s_0)[s_1] = (A \upharpoonright s_0)[s_0] - \{a\}$.

Suppose first that $(A \upharpoonright s_0)[s_1] = (A \upharpoonright s_0)[s_0] \cup \{a\}$. In this case, $a$ must have been added for the sake of some $\rho_0$-strategy at some stage between $s_0$ and $s_1$. This can only happen as part of an attempt to correct a value of the current version of $\rho_0$'s non-backup functional at stage $s_0$. Since $\rho_1$ acted at stage $s_0$, $\rho_0^\frown\langle\text{num}\rangle \subseteq \rho_1$. By section 3.2, cases I.2 or II.3A.2, $\rho_0$ will never allow $\rho_1$ to act again until $A(a)[s_0]$ is restored, unless there is a failure causing $a$ to be made available to the backup functional. (Recall that $\rho_0$ itself is not initialized by the failure of *ii.*) But at such a stage $\rho_0^\frown\langle\infty\rangle$ acts, initializing $\rho_0^\frown\langle\text{num}\rangle$, and $\rho_1$ as well. So *iii* holds.

Otherwise, suppose $(A \upharpoonright s_0)[s_1] = (A \upharpoonright s_0)[s_0] - \{a\}$. In this case, $a$ must have been removed for the sake of some $\rho_0$-strategy at some stage between $s_0$ and $s_1$. This can only happen as part of an attempt to correct a value of $\rho_0$'s backup functional. Since $\rho_1$ acted at stage $s$, $\rho_0^\frown\langle\infty\rangle \subseteq \rho_1$. However, in this case, $\rho_0$ would never allow $\rho_1$ to act until $A(a)[s_0]$ is restored under section 3.2, cases I.2 or II.3A.2, contradicting the failure of *i*. This establishes the lemma. □

To show that R is satisfied, we suppose that $\Phi(W_l^{W_e}) = A$ and $\Psi(A) = W_l^{W_e}$, since otherwise there is nothing to prove. Naturally, there are two possibilities, depending on which infinitary outcome of $\tau$ lies on the true path.

If $\tau^\frown\langle\text{num}\rangle \subset f$, then we may assume that after stage $s_0$, $f_s$ never branches back through $\tau^\frown\langle\infty\rangle$. Note that $\Gamma^\tau$ is never initialized after stage $s_0$. Recall that we assume every worker for $\tau$ on the true path has a finitary outcome. Since $\tau^\frown\langle\text{num}\rangle$ acts infinitely often, every link with top $\tau$ is eventually removed. Once a worker $\sigma$ for $\tau$ sets a link and this link is removed, either $\sigma$ succeeds and never acts again, or the node itself, and indeed, everything below the $\langle\text{num}\rangle$ outcome of its master, is initialized, under section 3.2, case II.3B. This means that every one of the infinitely many workers for $\tau$ along the true path has an opportunity to act after $s_0$, and, if it does act, its action succeeds. This shows $\Gamma^\tau(k) = K(k)$ whenever $\Gamma^\tau(k)\downarrow$. For each $k$, however, $\Gamma^\tau(k)$ must converge permanently, since otherwise the $k$th worker for $\tau$ along the true path would have outcome $\langle\infty\rangle$ infinitely often. This shows R is satisfied if $\tau^\frown\langle\text{num}\rangle \subset f$.

The argument in the case $\tau^\frown\langle\infty\rangle \subset f$ is a little more subtle. Eventually, $\gamma_\infty(k)$ is defined, since otherwise the $\sigma$ to which R$(k)$ is assigned must have an infinite outcome on the true path. This follows since, by Lemma 4.3, R$(k)$ never has to pick a new attacker after some point. Let $a$ be the attacker for R$(k)$. But, when $a$ is removed by the $\sigma$-strategy at some stage $s+1$ because $k$ has entered $K$, $A$ is then in the same state as it was before $a$ ever entered $A$, by Lemma 4.4, *iii.* At this stage, the $x \in W_l^{W_e}[s]$ which the backup strategy for keeping $\gamma_\infty^\tau(k)[s]$ correct is depending on was not yet an element of $W_l^{W_e}$. Thus, $\Psi(A;x)[s+1] = 0$. For this element to leave, $W_e$ must change on $\phi_l(W_e;x)[s] = \gamma_\infty^\tau(n)[s]$. Hence at the next $\tau$-expansionary stage, $\gamma^\tau(k) = \phi_l(W_e;x)[s]$ must diverge, and can then be reset correctly. This establishes the result.

# References


[1] Arslanov, M., LaForte, G., and Slaman, T., *Relative enumerability in the difference hierarchy,* to appear, J. Symbolic Logic.

[2] R. Downey, *The $0'''$ priority method with special attention to density results*, Recursion Theory Week: Proceedings, Oberwohlfach 1989 (K. Ambos-Spies, *et al.* , eds., Springer, Berlin, 1990).

[3] Sacks, G., *Recursive enumerability and the jump operator,* Trans. Am. Math. Soc. 108 (1963) 223-239.

[4] Shore, R., *A non-inversion theorem for the jump operator,* Ann. Pure and Appl. Logic 40 (1988) 277-303.

[5] Soare, R., *Recursively enumerable sets and degrees* (Springer, Berlin, 1987).